REMARKS

Claims 1, 3-16 and 18-21 were pending at the time of examination. Claims 1, 4-6, 10, 12-13, 16 and 19-21 have been amended. Claims 3, 11 and 18 have been canceled. New claim 22 has been added. No new matter has been added. The Applicants respectfully request reconsideration based on the foregoing amendments and these remarks.

Claim Rejections – 35 U.S.C. § 112

Claim 7 was rejected under 35 U.S.C § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the Applicants regard as the invention. In particular, the limitation "the source program" lacks antecedent basis. Claim 1, as amended, contains antecedent basis for the limitation "the source program," thereby making the rejection of claim 7 moot. The Applicants submit that claim 7 is now in proper order and that the rejection should be withdrawn.

Claim Rejections – 35 U.S.C. § 103

Claims 1, 7-8, 10-13, 16 and 21 were rejected under 35 U.S.C § 103(a) as being unpatentable over U.S. Patent No. 5,991,173 to Unger et al. (hereinafter "Unger") in view of U.S. Patent No. 6,163,811 to Porter (hereinafter "Porter"). The Applicants respectfully traverse this rejection.

Both Unger and Porter describe various methods of compression of data structures. Unger describes methods of compressing natural language text, while Porter describes token-based source file compression. However, neither Unger nor Porter reasonably suggest or describe the subject matter of amended claim 1, as will be discussed below.

Claim 1 recites the steps of:

"encoding a program symbol name to produce an encoded program symbol name, without changing the non-program symbol information;

generating a differential name for the encoded program symbol name relative to a base symbol for the program symbol, the differential name having a reduced-size format as compared to the encoded program symbol name; and

producing a compressed compiler product including the generated differential name."

The Examiner has asserted that Unger discloses that the compiler information being compressed includes textual symbol names. The Applicants respectfully disagree. Unger's method of compression compares the words (including *numeric strings, decimal points, currency* *symbols*, etc.) to a predetermined dictionary, and then to a supplemental dictionary if the words cannot be found in the first dictionary (*see* col. 10, lines 23-39). Numeric characters can be encoded with a special predetermined "numeric" dictionary (*see id.*). As such, the symbols described in Unger are merely characters in a text file and have no function or meaning outside of the context of the text file being compressed.

In contrast, claim 1 requires encoding "program symbol names" <u>without changing the non-program symbol information</u>. As found in the Specification at page 2,

> Source programs typically use a sequence of tokens to name program symbols. The portion 102 illustrated in FIG. 1 declares three program symbols, a "namespace", an "int" variable, and a "long" variable. The "namespace" is a container, and has the two variables as its members. Typically, programs use short context-dependent names for these symbols.

Thus, the program symbol names are not simply characters in a text file as taught by Unger, but are more correctly characterized as programming references or pointers with meaning outside of the context of a text file as taught by Unger. That is, a given sequence of characters forming a program symbol name can have varying differential names, depending on the context of the program symbol name. Therefore, when a differential name is generated for the encoded program symbol, the differential name is generated <u>relative to a base symbol</u> (e.g., a class or other type of container object), as described in claim 1, so that the program symbol name and its context can be uniquely identified. Furthermore, in Unger, the goal is to encode as much as possible of the text file, while in the Applicants' invention, only the program symbol names are encoded and the non-program symbol information is left unaltered.

Neither Unger nor Porter, alone or in combination, anticipate or render claim 1 obvious. As was discussed above, Unger does not describe compression of encoded program symbol names by using differential names. Porter does not cure this deficiency of Unger, because Porter describes only compression of entire source files over a distributed network. Furthermore, while Porter describes creating a symbol table for operands (i.e. "=", "+", "-", etc.) and their substitution (*see* col. 6, ll. 17-27; FIG. 3a), Porter does not describe encoded program symbol names or differential names as contemplated by the present claim. For at least the above reasons, the rejection of claim 1 is unsupported by the cited art and should be withdrawn.

Claim 16 is a *Beauregard* claim corresponding to claim 1. For reasons substantially similar to those set forth above, the Applicants respectfully contend that the rejection of claim 16 is unsupported by the cited art and should be withdrawn.

Claims 4-9 depend from claim 1, and the rejections of these claims are unsupported by the cited art for at least the reasons discussed above with regards to claim 1, and should be withdrawn.

Claims 19-20 depend from claim 16, and the rejections of these claims are unsupported by the cited art for at least the reasons discussed above with regards to claim 16, and should be withdrawn.

Claim 10 describes a method for generating encoded program symbol names in an uncompressed form, and was rejected for the same rationale that was set forth in the rejection of claim 1. Claim 10, as amended, contains limitations relating to program symbol names, base symbols, and differential program symbol names and formats. For at least the reasons discussed above with regards to claim 1, the Applicants respectfully contend that the rejection of claim 10 is unsupported by the cited art and should be withdrawn.

Claim 12 depends from claim 10, and the rejection of this claim is therefore unsupported by the cited art for at least the same reasons, and should be withdrawn.

Claim 21 is a *Beauregard* claim corresponding to claim 1. For reasons substantially similar to those set forth above, the Applicants respectfully contend that the rejection of claim 16 is unsupported by the cited art and should be withdrawn.

Claim 13 was rejected for substantially the same reasons as claim 1. Claim 13 has been amended to include the limitation that the enhanced compiler includes "one or more differential names corresponding to the program symbol names." The program symbol names and the differential names have been discussed above with respect to the rejection of claim 1. For reasons substantially similar to those set forth above with regards to claim 1, the Applicants respectfully contend that the rejection of claim 13 is unsupported by the cited art and should be withdrawn.

Claims 14 and 15 depend from claim 13, and the rejections of these claims are unsupported by the cited art for at least the reasons discussed with regards to claim 13, and should be withdrawn

Added claims

Claim 22 has been added to further define features of the invention. No new matter has been added. Claim 22 depends from claim 1, and is therefore allowable for at least the reasons discussed above.

Conclusion

The Applicants believe that all pending claims are allowable and respectfully request a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

Fredrik Mollborn

Fredrik Mollborn
Reg. No. 48,587

P.O. Box 778
Berkeley, CA 94704-0778
(650) 961-8300